Lecture 10: Exploration

CS234: RL

Emma Brunskill

Spring 2017

With thanks to Christoph Dann some slides on PAC vs regret vs PAC-uniform

Today

- Review: Importance of exploration in RL
- Performance criteria
- Optimism under uncertainty
  - Review of UCRL2
  - Rmax
- Scaling up (generalization + exploration)


4
148400

Montezuma’s Revenge

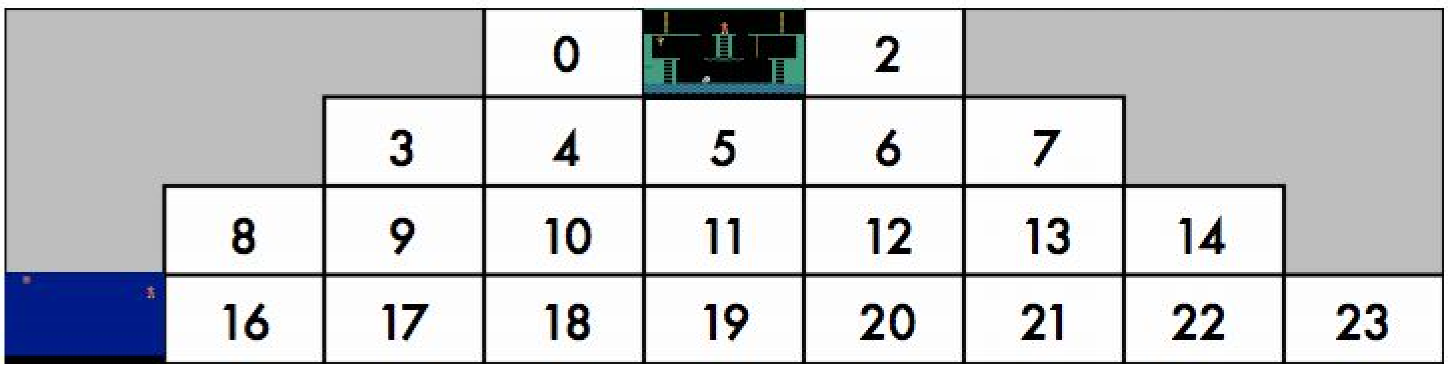
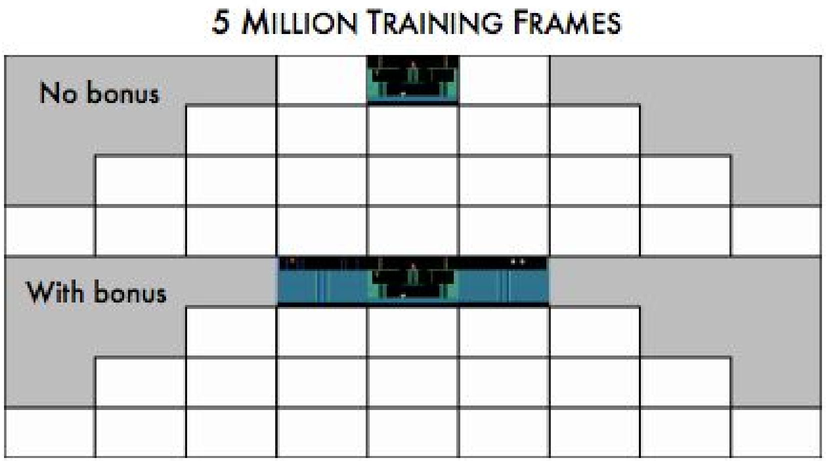


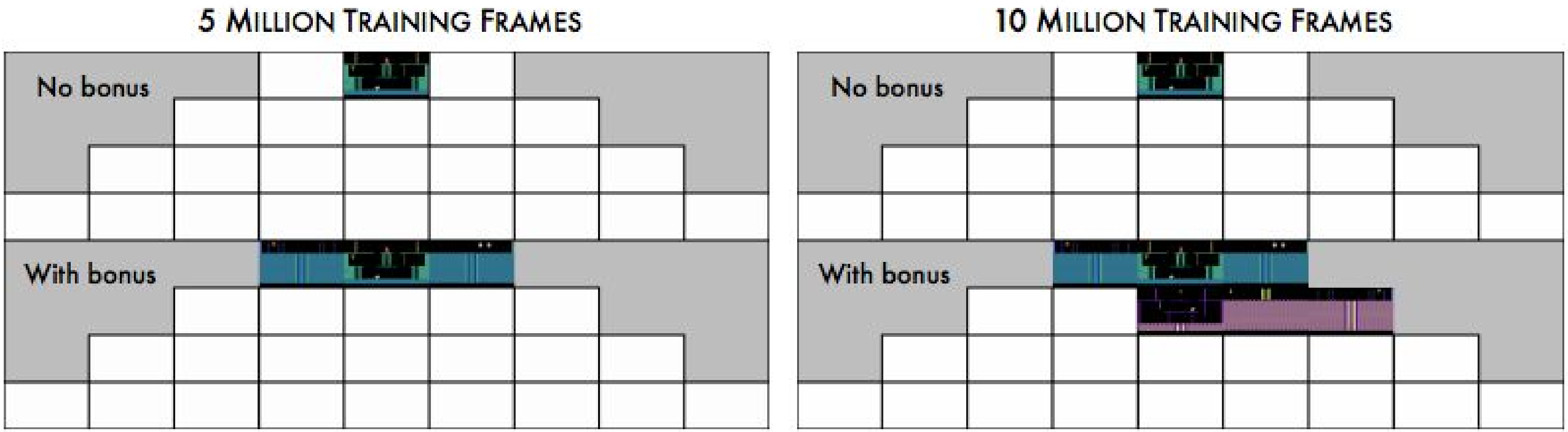
Figure 7: Layout of levels in MONTEZUMA’S REVENGE, with rooms numbered from 0 to 23. The agent begins in room 1 and completes the level upon reaching room 15 (depicted).

Systematic Exploration Key



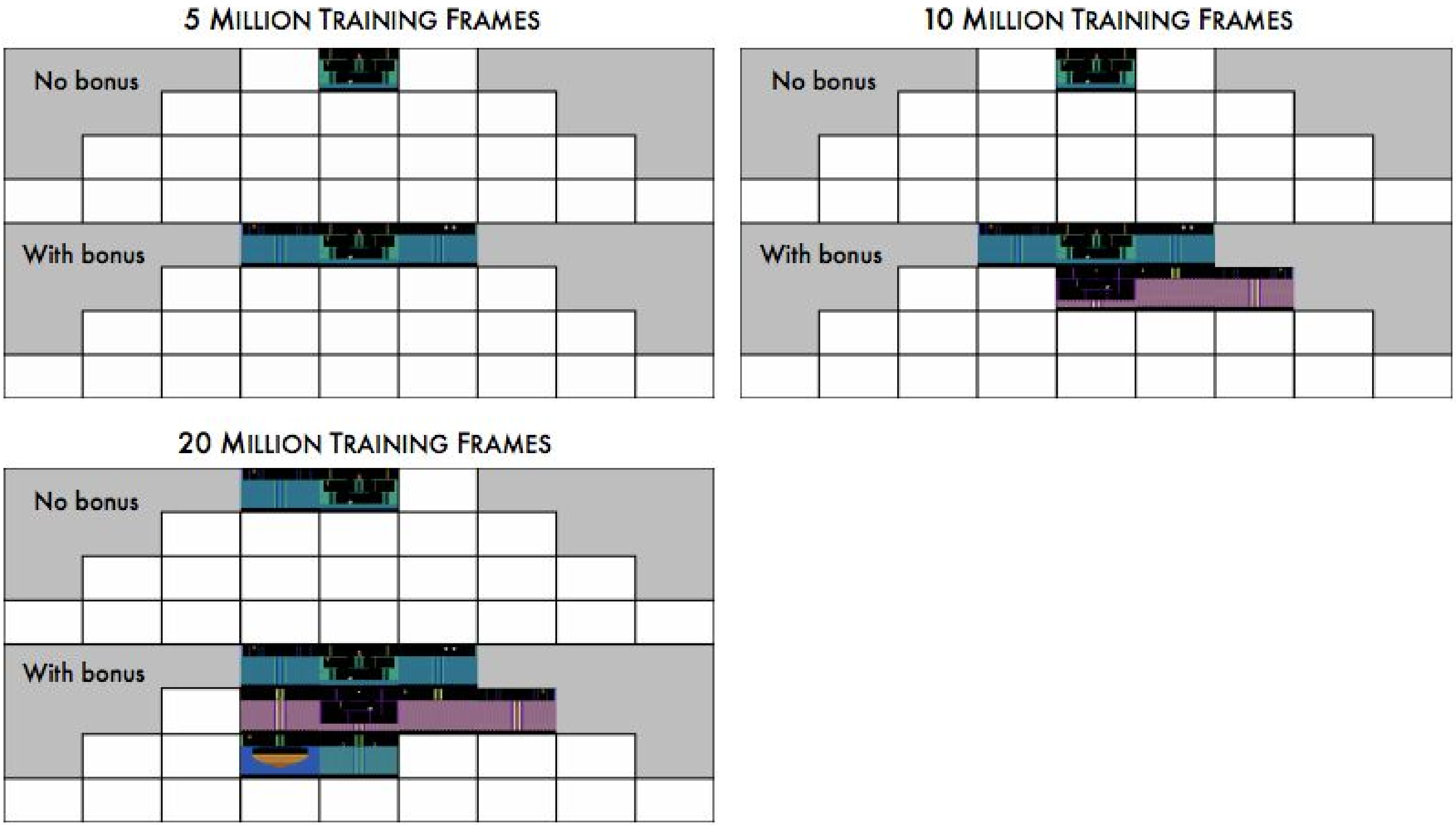
[Unifying Count-Based Exploration and Intrinsic Motivation, https://arxiv.org/pdf/1606.01868.pdf](https://arxiv.org/pdf/1606.01868.pdf)

Systematic Exploration Key



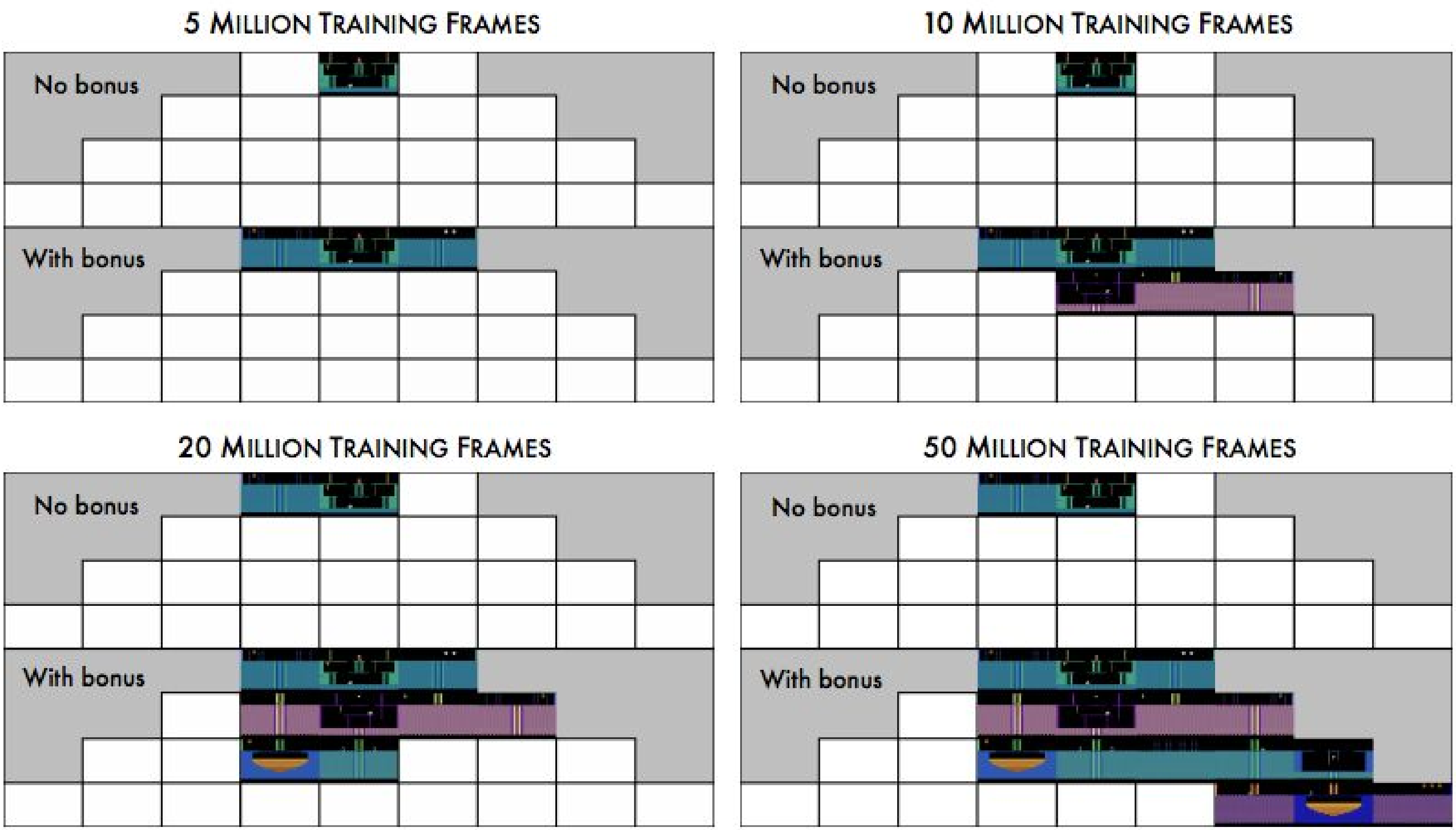
[Unifying Count-Based Exploration and Intrinsic Motivation, https://arxiv.org/pdf/1606.01868.pdf](https://arxiv.org/pdf/1606.01868.pdf)

Systematic Exploration Key



[Unifying Count-Based Exploration and Intrinsic Motivation, https://arxiv.org/pdf/1606.01868.pdf](https://arxiv.org/pdf/1606.01868.pdf)

Systematic Exploration Key



[Unifying Count-Based Exploration and Intrinsic Motivation, https://arxiv.org/pdf/1606.01868.pdf](https://arxiv.org/pdf/1606.01868.pdf)

Systematic Exploration Important

Intelligent Tutoring

[e.g.Mandel, Liu, Brunskill, Popovic '14]

Adaptive Treatment

[Guez et al '08]

- In Montezuma's revenge, data = computation
- In many applications, data = people
    - Data = interactions with a student / patient / customer ...
- Need sample efficient RL = need careful exploration

Performance of RL Algorithms

- Convergence
- Asymptotically optimal
- Probably approximately correct
- Minimize / sublinear regret

Last Lecture: UCRL2

Near-optimal Regret Bounds for Reinforcement Learning

1. Given past experience data D, for each (s,a) pair
   - Construct a confidence set over possible transition model
   - Construct a confidence interval over possible reward
2. Compute policy and value by being optimistic with respect to these sets
3. Execute resulting policy for a particular number of steps

UCLR2

- Strong regret bounds

$$\Delta(M, \mathfrak{A}, s, T) := T\rho^*(M) - R(M, \mathfrak{A}, s, T)$$

$$\Delta(M, \textsc{Ucrl2}, s, T) \leq 34 \cdot DS\sqrt{AT \log\left(\tfrac{T}{\delta}\right)}$$

D = diameter
A = number of actions
T = number of time steps algorithm acts for
M = MDP
s = a particular state
S = size of state space
delta = high probability?

UCRL2: Optimistic Under Uncertainty

1. Given past experience data D, for each (s,a) pair
   - Construct a confidence set over possible transition model
   - Construct a confidence interval over possible reward
2. Compute policy and value by being optimistic with respect to these sets
3. Execute resulting policy for a particular number of steps

Optimism under Uncertainty

- Consider the set D of (s,a,r,s') tuples observed so far
  - Could be zero set (no experience yet)
- Assume real world is a particular MDP M1
  - M1 generated observed data D
- If knew M1, just compute optimal policy for M1
  - and will achieve high reward
- But many MDPs could have generated D
- Given this uncertainty (over true world models) act optimistically

Optimism under Uncertainty

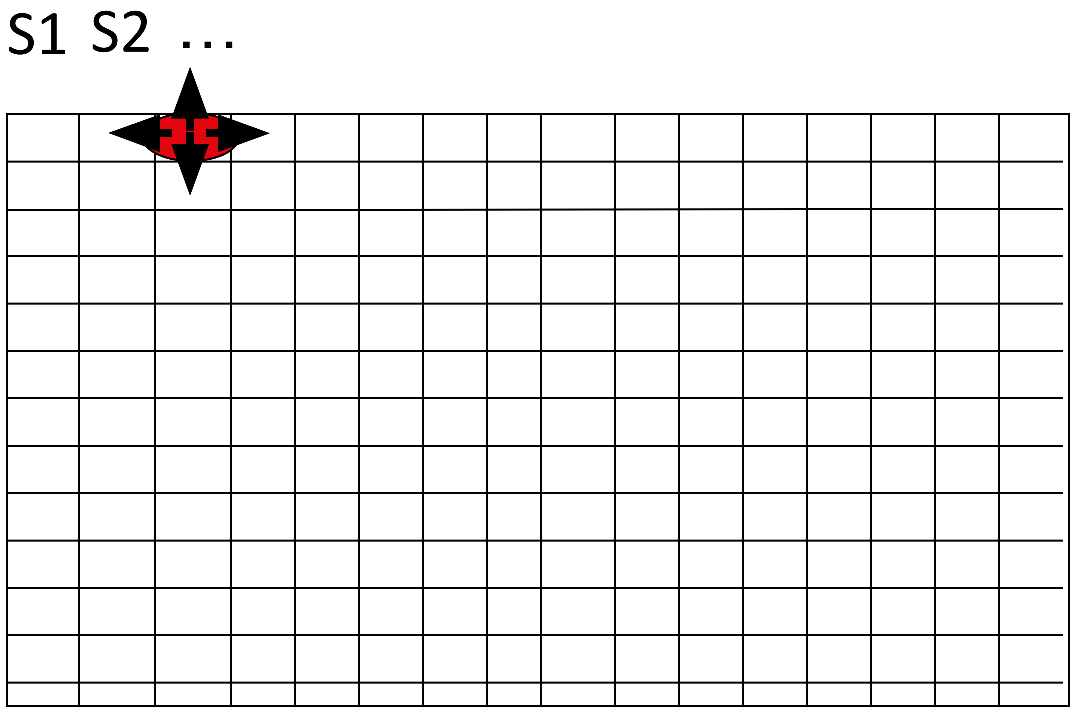
- Why is this powerful?
  - Either
    - Hypothesized optimism is empirically valid (world really is as wonderful as dream it is)
      $\longrightarrow$ Gather high reward
    - or, World isn't that good (lower rewards than expected)
      $\longrightarrow$ Learned something. Reduced uncertainty over how
      the world works.

Optimism under Uncertainty

- Used in many algorithms that are PAC or regret
- Last lecture: UCRL2
  - Continuous representation of uncertainty
  - Confidence sets over model parameters
  - Regret bounds
- Today: R-max (Brafman and Tenneholtz)
  - Discrete representation of uncertainty
  - Probably Approximately Correct bounds

R-max (Brafman & Tennenholtz)

http://www.jmlr.org/papers/v3/brafman02a.html

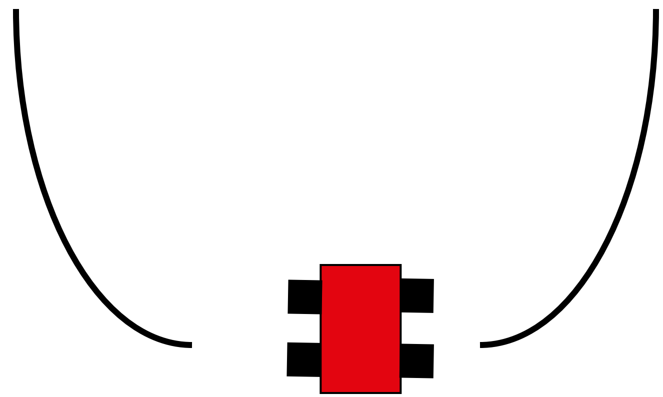


Example domain

- Discrete set of states and actions
- Want to maximize discounted sum of rewards

R-max is Model-based RL

Use data to construct transition and reward models
& compute policy (e.g. using value iteration)



Act in world

Rmax leverages optimism under uncertainty!

R-max Algorithm:
Initialize: Set all (s,a) to be “Unknown”

Known/
Unknown

	S1	S2	S3	S4	…
↑	U	U	U	U	
→	U	U	U	U	
↓	U	U	U	U	
←	U	U	U	U	

R-max Algorithm:
Initialize: Set all (s,a) to be “Unknown”

Known/
Unknown

	S1	S2	S3	S4	…
↑	U	U	U	U	
→	U	U	U	U	
↓	U	U	U	U	
←	U	U	U	U	

In the “known” MDP, any unknown (s,a) pair has its dynamics set as a self loop & reward = Rmax

R-max Algorithm: Creates a “Known” MDP

Reward

Known/ Unknown

	S1	S2	S3	S4	…
↑	U	U	U	U	
→	U	U	U	U	
↓	U	U	U	U	
←	U	U	U	U	

	S1	S2	S3	S4	…
↑	R_{max}	R_{max}	R_{max}	R_{max}	
→	R_{max}	R_{max}	R_{max}	R_{max}	
↓	R_{max}	R_{max}	R_{max}	R_{max}	
←	R_{max}	R_{max}	R_{max}	R_{max}	

Transition Counts

	S1	S2	S3	S4	…
↑	0	0	0	0	
→	0	0	0	0	
↓	0	0	0	0	
←	0	0	0	0	

In the “known” MDP, any unknown (s,a) pair has its dynamics set as a self loop & reward = Rmax

R-max Algorithm

Plan in known MDP

R-max: Planning

- Compute optimal policy π_{known} for "known" MDP

Exercise: What Will Initial Value of Q(s,a) be for each (s,a) Pair in the Known MDP? What is the Policy?

Known/ Unknown

	S1	S2	S3	S4	…
↑	U	U	U	U	
→	U	U	U	U	
↓	U	U	U	U	
←	U	U	U	U	

Reward

	S1	S2	S3	S4	…
↑	R_{max}	R_{max}	R_{max}	R_{max}	
→	R_{max}	R_{max}	R_{max}	R_{max}	
↓	R_{max}	R_{max}	R_{max}	R_{max}	
←	R_{max}	R_{max}	R_{max}	R_{max}	

Transition Counts

	S1	S2	S3	S4	…
↑	0	0	0	0	
→	0	0	0	0	
↓	0	0	0	0	
←	0	0	0	0	

In the “known” MDP, any unknown (s,a) pair has its dynamics set as a self loop & reward = Rmax

R-max Algorithm

Act using policy

Plan in known MDP

- Given optimal policy π_{known} for “known” MDP
- Take best action for current state $\pi_{known}(s)$, transition to new state s’ and get reward r

R-max Algorithm

Act using policy

Plan in known MDP

Update state-action counts

Update Known MDP Given Recent (s,a,r,s')

Known/
Unknown

	S2	S2	S3	S4	…
↑	U	U	U	U	
→	U	U	U	U	
↓	U	U	U	U	
←	U	U	U	U	

Reward

	S2	S2	S3	S4	…
↑	R_{max}	R_{max}	R_{max}	R_{max}	
→	R_{max}	R_{max}	R_{max}	R_{max}	
↓	R_{max}	R_{max}	R_{max}	R_{max}	
←	R_{max}	R_{max}	R_{max}	R_{max}	

Transition
Counts

	S2	S2	S3	S4	…
↑	0	0	0	0	
→	0	0	1	0	
↓	0	0	0	0	
←	0	0	0	0	

Increment counts for
state-action tuple

Update Known MDP

Known/ Unknown

	S2	S2	S3	S4	…
↑	U	U	U	U	
→	U	U	**K**	U	
↓	U	U	U	U	
←	U	U	U	U	

Reward

	S2	S2	S3	S4	…
↑	R_{max}	R_{max}	R_{max}	R_{max}	
→	R_{max}	R_{max}	**R**	R_{max}	
↓	R_{max}	R_{max}	R_{max}	R_{max}	
←	R_{max}	R_{max}	R_{max}	R_{max}	

Transition Counts

	S2	S2	S3	S4	…
↑	3	3	4	3	
→	2	4	**5**	0	
↓	4	0	4	4	
←	2	2	4	1	

If counts for (s,a) > N, (s,a) becomes known: **use observed data to estimate transition & reward model** for (s,a) when planning

Estimate Models for Known (s,a) Pairs

- Use maximum likelihood estimates
- Transition model estimation

  $P(s' | s,a) = \text{counts}(s,a \longrightarrow s') \,/\, \text{counts}(s,a)$

- Reward model estimation

$R(s,a) = \sum \text{observed rewards } (s,a) \,/\, \text{counts}(s,a)$

where $\text{counts}(s,a) = \#$ of times observed (s,a)

When Does Policy Change When a (s,a) Pair Becomes Known?

Reward

Known/ Unknown

	S2	S2	S3	S4	…
↑	U	U	U	U	
→	U	U	**K**	U	
↓	U	U	U	U	
←	U	U	U	U	

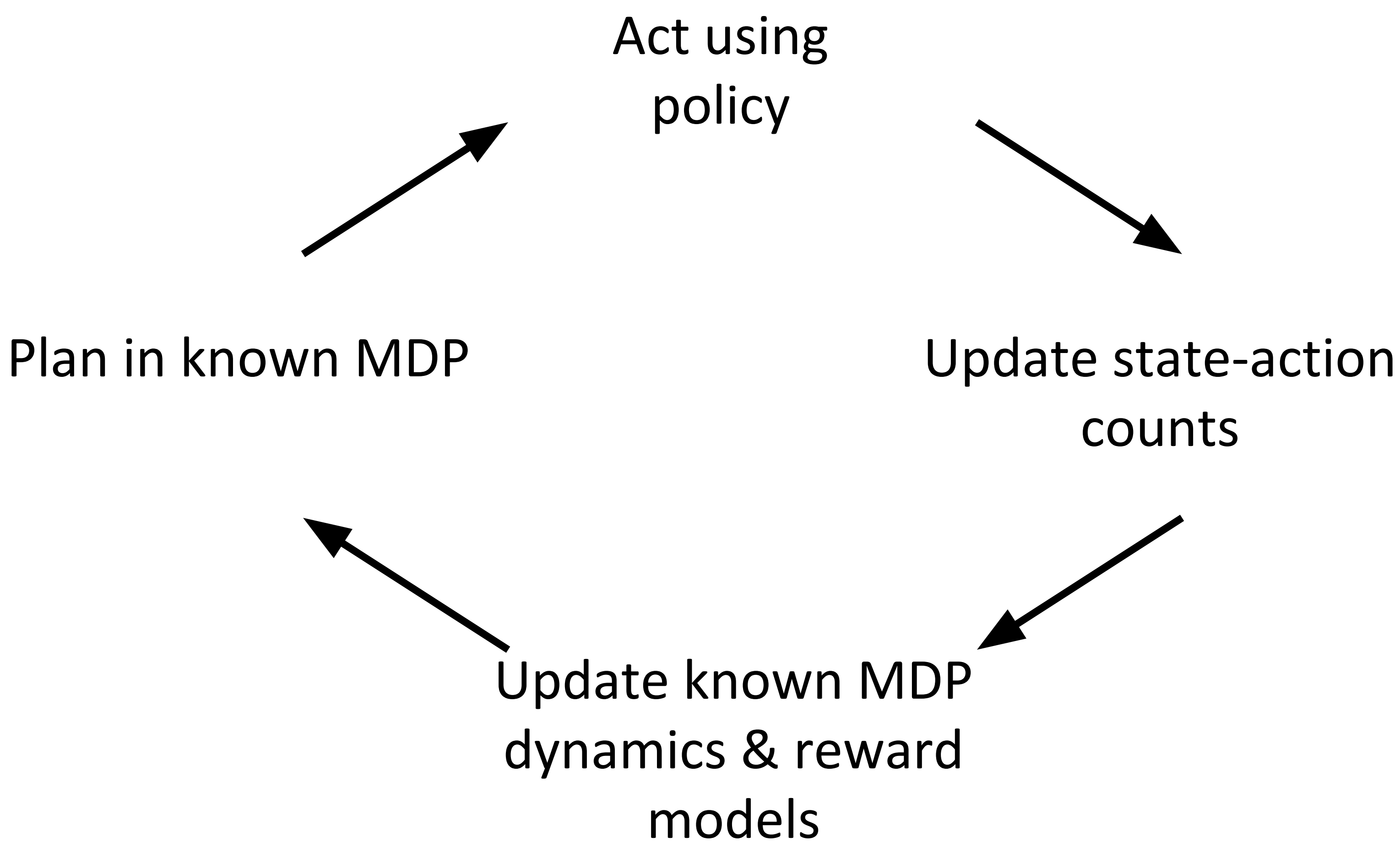
	S2	S2	S3	S4	…
↑	R_{max}	R_{max}	R_{max}	R_{max}	
→	R_{max}	R_{max}	**R**	R_{max}	
↓	R_{max}	R_{max}	R_{max}	R_{max}	
←	R_{max}	R_{max}	R_{max}	R_{max}	

Transition Counts

	S2	S2	S3	S4	…
↑	3	3	4	3	
→	2	4	**5**	0	
↓	4	0	4	4	
←	2	2	4	1	

If counts for (s,a) > N, (s,a) becomes known: **use observed data to estimate transition & reward model** for (s,a) when planning

R-max Algorithm



R-max and Optimism Under Uncertainty

- UCRL2 used a continuous measure of uncertainty
  - Confidence intervals over model parameters
- R-max uses a hard threshold: binary uncertainty
  - Either have enough information to rely on empirical estimates
  - Or don’t (and if don’t, be optimistic)

R-max (Brafman and Tennenholtz).

Slight modification of R-max (Algorithm 1) pseudo code in [Reinforcement Learning in Finite MDPs: PAC Analysis](#) (Strehl, Li, LIttman 2009)

Inputs: $S, A, \gamma, m, \varepsilon_1$, and $U(\cdot,\cdot)$
for all (s,a) **do**
    $Q(s,a) \leftarrow$ $\boxed{R_{max} / (1-\gamma)}$
    $r(s,a) \leftarrow 0$
    $n(s,a) \leftarrow 0$
    for all $s' \in S$ **do**
        $n(s,a,s') \leftarrow 0$
    end for
end for
for $t = 1,2,3,\cdots$ **do**
    Let s denote the state at time t.
    Choose action $a := \text{argmax}_{a' \in A} Q(s,a')$.
    Let r be the immediate reward and s' the next state after executing action a from state s.
    if $n(s,a) < m$ **then**
        $n(s,a) \leftarrow n(s,a) + 1$
        $r(s,a) \leftarrow r(s,a) + r$ *// Record immediate reward*
        $n(s,a,s') \leftarrow n(s,a,s') + 1$ *// Record immediate next-state*
        if $n(s,a) = m$ **then**
            for $i = 1,2,3,\cdots,\left\lceil \frac{\ln(1/(\varepsilon_1(1-\gamma)))}{1-\gamma} \right\rceil$ **do**
                for all $(\bar{s},\bar{a})$ **do**
                    if $n(\bar{s},\bar{a}) \geq m$ **then**
                        $Q(\bar{s},\bar{a}) \leftarrow \hat{R}(\bar{s},\bar{a}) + \gamma \sum_{s'} \hat{T}(s'|\bar{s},\bar{a}) \max_{a'} Q(s',a')$.
                    end if
                end for
            end for
        end if
    end if
end for

Reminder:
Probably Approximately Correct RL

Definition 2 *An algorithm $\mathcal{A}$ is said to be an* **efficient PAC-MDP** *(Probably Approximately Correct in Markov Decision Processes) algorithm if, for any $\varepsilon > 0$ and $0 < \delta < 1$, the per-timestep computational complexity, space complexity, and the sample complexity of $\mathcal{A}$ are less than some polynomial in the relevant quantities $(S, A, 1/\varepsilon, 1/\delta, 1/(1-\gamma))$, with probability at least $1-\delta$. It is simply* **PAC-MDP** *if we relax the definition to have no computational complexity requirement.*

See e.g. [Reinforcement Learning in Finite MDPs: PAC Analysis](http://www.jmlr.org/papers/volume10/strehl09a/strehl09a.pdf) (Strehl, Li, LIttman 2009, http://www.jmlr.org/papers/volume10/strehl09a/strehl09a.pdf)

R-max is a Probably Approximately Correct RL Algorithm

On all but the following number of steps, chooses action whose value is at least epsilon-close to V* with probability at least 1-delta

$$\underbrace{\tilde{O}}_{\text{ignore log factors}}(S^2 A / (\varepsilon^3 (1-\gamma)^6))$$

For proof see
original R-max paper, http://www.jmlr.org/papers/v3/brafman02a.html
or Reinforcement Learning in Finite MDPs: PAC Analysis (Strehl, Li, LIttman 2009, http://www.jmlr.org/papers/volume10/strehl09a/strehl09a.pdf)

Sufficient Condition for PAC Model-based RL

(see Strehl, Li, LIttman 2009, http://www.jmlr.org/papers/volume10/strehl09a/strehl09a.pdf)

Theorem 10 *Let $\mathcal{A}(\varepsilon,\delta)$ be any greedy learning algorithm such that, for every timestep t, there exists a set K_t of state-action pairs that depends only on the agent's history up to timestep t. We assume that $K_t = K_{t+1}$ unless, during timestep t, an update to some state-action value occurs or the escape event A_K happens. Let M_{K_t} be the known state-action MDP and π_t be the current greedy policy, that is, for all states s, $\pi_t(s) = \mathrm{argmax}_a Q_t(s,a)$. Furthermore, assume $Q_t(s,a) \leq V_{\max}$ for all t and (s,a). Suppose that for any inputs ε and δ, with probability at least $1-\delta$, the following conditions hold for all states s, actions a, and timesteps t: (1) $V_t(s) \geq V^*(s) - \varepsilon$ (optimism), (2) $V_t(s) - V^{\pi_t}_{M_{K_t}}(s) \leq \varepsilon$ (accuracy), and (3) the total number of updates of action-value estimates plus the number of times the escape event from K_t, A_K, can occur is bounded by $\zeta(\varepsilon,\delta)$ (learning complexity). Then, when $\mathcal{A}(\varepsilon,\delta)$ is executed on any MDP M, it will follow a 4ε-optimal policy from its current state on all but*

$$O\left(\frac{V_{\max}\zeta(\varepsilon,\delta)}{\varepsilon(1-\gamma)} \ln\frac{1}{\delta} \ln\frac{1}{\varepsilon(1-\gamma)}\right)$$

timesteps, with probability at least $1-2\delta$.

Sufficient Condition for PAC Model-based RL

(see Strehl, Li, LIttman 2009, http://www.jmlr.org/papers/volume10/strehl09a/strehl09a.pdf)

Theorem 10 $\boxed{\textit{Let } \mathcal{A}(\varepsilon,\delta) \textit{ be any greedy learning algorithm}}$ *such that, for every timestep t, there exists a set K_t of state-action pairs that depends only on the agent's history up to timestep t. We assume that $K_t = K_{t+1}$ unless, during timestep t, an update to some state-action value occurs or the escape event A_K happens. Let M_{K_t} be the known state-action MDP and π_t be the current greedy policy, that is, for all states s, $\pi_t(s) = \operatorname{argmax}_a Q_t(s,a)$. Furthermore, assume $Q_t(s,a) \le V_{\max}$ for all t and (s,a). Suppose that for any inputs ε and δ, with probability at least $1-\delta$, the following conditions hold for all states s, actions a, and timesteps t: (1) $V_t(s) \ge V^*(s) - \varepsilon$ (optimism), (2) $V_t(s) - V^{\pi_t}_{M_{K_t}}(s) \le \varepsilon$ (accuracy), and (3) the total number of updates of action-value estimates plus the number of times the escape event from K_t, A_K, can occur is bounded by $\zeta(\varepsilon,\delta)$ (learning complexity). Then, when $\mathcal{A}(\varepsilon,\delta)$ is executed on any MDP M, it will follow a 4ε-optimal policy from its current state on all but*

$$O\left(\frac{V_{\max}\zeta(\varepsilon,\delta)}{\varepsilon(1-\gamma)} \ln\frac{1}{\delta} \ln\frac{1}{\varepsilon(1-\gamma)}\right)$$

timesteps, with probability at least $1-2\delta$.

- Greedy learning algorithm here means that maintains Q estimates and for a particular state s chooses action a = argmax Q(s,a)
- Note: not saying yet how construct these Q!

Sufficient Condition for PAC Model-based RL

(see Strehl, Li, LIttman 2009, http://www.jmlr.org/papers/volume10/strehl09a/strehl09a.pdf)

Theorem 10 *Let $\mathcal{A}(\varepsilon,\delta)$ be any greedy learning algorithm such that, for every timestep t, there exists a set K_t of state-action pairs that depends only on the agent's history up to timestep t. We assume that $K_t = K_{t+1}$ unless, during timestep t, an update to some state-action value occurs or the escape event A_K happens. Let M_{K_t} be the known state-action MDP and π_t be the current greedy policy, that is, for all states s, $\pi_t(s) = \text{argmax}_a Q_t(s,a)$. Furthermore, assume $Q_t(s,a) \leq V_{\max}$ for all t and (s,a). Suppose that for any inputs ε and δ, with probability at least $1-\delta$, the following conditions hold for all states s, actions a, and timesteps t: (1) $V_t(s) \geq V^*(s) - \varepsilon$ (optimism), (2) $V_t(s) - V^{\pi_t}_{M_{K_t}}(s) \leq \varepsilon$ (accuracy), and (3) the total number of updates of action-value estimates plus the number of times the escape event from K_t, A_K, can occur is bounded by $\zeta(\varepsilon,\delta)$ (learning complexity). Then, when $\mathcal{A}(\varepsilon,\delta)$ is executed on any MDP M, it will follow a 4ε-optimal policy from its current state on all but*

$$O\left(\frac{V_{\max}\zeta(\varepsilon,\delta)}{\varepsilon(1-\gamma)} \ln\frac{1}{\delta} \ln\frac{1}{\varepsilon(1-\gamma)}\right)$$

timesteps, with probability at least $1-2\delta$.

- For example, K_t = known set of (s,a) pairs in R-max algorithm at time step t

Sufficient Condition for PAC Model-based RL

(see Strehl, Li, LIttman 2009, http://www.jmlr.org/papers/volume10/strehl09a/strehl09a.pdf)

Theorem 10 *Let $\mathcal{A}(\varepsilon, \delta)$ be any greedy learning algorithm such that, for every timestep t, there exists a set K_t of state-action pairs that depends only on the agent's history up to timestep t. We assume that $K_t = K_{t+1}$ unless, during timestep t, an update to some state-action value occurs or the escape event A_K happens. Let M_{K_t} be the known state-action MDP and π_t be the current greedy policy, that is, for all states s, $\pi_t(s) = \operatorname{argmax}_a Q_t(s,a)$. Furthermore, assume $Q_t(s,a) \leq V_{\max}$ for all t and (s,a). Suppose that for any inputs ε and δ, with probability at least $1-\delta$, the following conditions hold for all states s, actions a, and timesteps t: (1) $V_t(s) \geq V^*(s) - \varepsilon$ (optimism), (2) $V_t(s) - V^{\pi_t}_{M_{K_t}}(s) \leq \varepsilon$ (accuracy), and (3) the total number of updates of action-value estimates plus the number of times the escape event from K_t, A_K, can occur is bounded by $\zeta(\varepsilon, \delta)$ (learning complexity). Then, when $\mathcal{A}(\varepsilon, \delta)$ is executed on any MDP M, it will follow a 4ε-optimal policy from its current state on all but*

$$O\left(\frac{V_{\max}\zeta(\varepsilon,\delta)}{\varepsilon(1-\gamma)} \ln\frac{1}{\delta} \ln\frac{1}{\varepsilon(1-\gamma)}\right)$$

timesteps, with probability at least $1-2\delta$.

- Choose to update estimate of Q values
  - Limiting number of updates of Q is slightly strange*
- or see escape event A_K = visit (s,a) pair not in K_t

Known State-Action MDP: Slightly Different than Rmax

Theorem 10 *Let $\mathcal{A}(\varepsilon,\delta)$ be any greedy learning algorithm such that, for every timestep t, there exists a set K_t of state-action pairs that depends only on the agent's history up to timestep t. We assume that $K_t = K_{t+1}$ unless, during timestep t, an update to some state-action value occurs or the escape event A_K happens. Let $\boxed{M_{K_t} \textit{ be the known state-action MDP}}$ and π_t be the current greedy*

- Assume there is some real MDP M (real world MDP)
- Given as input a ~Q(s,a) function for all (s,a)
  - For R-max algorithm ~Q(s,a) = R_{max} / (1-γ)

Known State-Action MDP: Slightly Different than Rmax

Theorem 10 *Let $\mathcal{A}(\varepsilon,\delta)$ be any greedy learning algorithm such that, for every timestep t, there exists a set K_t of state-action pairs that depends only on the agent's history up to timestep t. We assume that $K_t = K_{t+1}$ unless, during timestep t, an update to some state-action value occurs or the escape event A_K happens. Let M_{K_t} be the known state-action MDP and π_t be the current greedy*

- Assume there is some real MDP M (real world MDP)
- Given as input a ~Q(s,a) function for all (s,a)
  - For R-max algorithm ~Q(s,a) = R_{max} / (1-γ)
- Define M_{Kt} as follows
- Same action space as M, State space is same + s0
- s0 has 0 reward and all actions return it to itself (self looping)

Known State-Action MDP: Slightly Different than Rmax

Theorem 10 *Let $\mathcal{A}(\varepsilon,\delta)$ be any greedy learning algorithm such that, for every timestep t, there exists a set K_t of state-action pairs that depends only on the agent's history up to timestep t. We assume that $K_t = K_{t+1}$ unless, during timestep t, an update to some state-action value occurs or the escape event A_K happens. Let M_{K_t} be the known state-action MDP and π_t be the current greedy*

- Assume there is some real MDP M (real world MDP)
- Given as input a ~Q(s,a) function for all (s,a)
  - For R-max algorithm ~Q(s,a) = R_{max} / (1-γ)
- Define M_{Kt} as follows
- Same action space as M, State space is same + s0
- s0 has 0 reward and all actions return it to itself (self looping)
- For (s,a) pairs in K_t
  - Set transition and reward models to be same as real MDP M
  - **Not** the empirical estimate of the models!

Known State-Action MDP: Slightly Different than Rmax

Theorem 10 *Let $\mathcal{A}(\varepsilon,\delta)$ be any greedy learning algorithm such that, for every timestep t, there exists a set K_t of state-action pairs that depends only on the agent's history up to timestep t. We assume that $K_t = K_{t+1}$ unless, during timestep t, an update to some state-action value occurs or the escape event A_K happens. Let $\boxed{M_{K_t} \text{ be the known state-action MDP}}$ and π_t be the current greedy*

- Assume there is some real MDP M (real world MDP)
- Given as input a ~Q(s,a) function for all (s,a)
  - For R-max algorithm ~Q(s,a) = R_{max} / (1-γ)
- Define M_{Kt} as follows
- Same action space as M, State space is same + s0
- s0 has 0 reward and all actions return it to itself (self looping)
- For (s,a) pairs in K_t
  - Set transition and reward models to be same as real MDP M
  - **Not** the empirical estimate of the models!
- For (s,a) pairs not in K_t
  - Set R(s,a) = ~Q(s,a) and p(s0|s,a) = 1 (e.g. transition to s0)

Greedy Policy wrt however construct Q_t

(see Strehl, Li, LIttman 2009, http://www.jmlr.org/papers/volume10/strehl09a/strehl09a.pdf)

Theorem 10 *Let $\mathcal{A}(\varepsilon,\delta)$ be any greedy learning algorithm such that, for every timestep t, there exists a set K_t of state-action pairs that depends only on the agent's history up to timestep t. We assume that $K_t = K_{t+1}$ unless, during timestep t, an update to some state-action value occurs or the escape event A_K happens. Let M_{K_t} be the known state-action MDP and π_t be the current greedy policy, that is, for all states s, $\pi_t(s) = \mathrm{argmax}_a Q_t(s,a)$. Furthermore, assume $Q_t(s,a) \leq V_{\max}$ for all t and (s,a). Suppose that for any inputs ε and δ, with probability at least $1-\delta$, the following conditions hold for all states s, actions a, and timesteps t: (1) $V_t(s) \geq V^*(s) - \varepsilon$ (optimism), (2) $V_t(s) - V^{\pi_t}_{M_{K_t}}(s) \leq \varepsilon$ (accuracy), and (3) the total number of updates of action-value estimates plus the number of times the escape event from K_t, A_K, can occur is bounded by $\zeta(\varepsilon,\delta)$ (learning complexity). Then, when $\mathcal{A}(\varepsilon,\delta)$ is executed on any MDP M, it will follow a 4ε-optimal policy from its current state on all but*

$$O\left(\frac{V_{\max}\zeta(\varepsilon,\delta)}{\varepsilon(1-\gamma)}\ln\frac{1}{\delta}\ln\frac{1}{\varepsilon(1-\gamma)}\right)$$

timesteps, with probability at least $1-2\delta$.

Q_t Values Always Upper Bounded

Theorem 10 *Let $\mathcal{A}(\varepsilon,\delta)$ be any greedy learning algorithm such that, for every timestep t, there exists a set K_t of state-action pairs that depends only on the agent's history up to timestep t. We assume that $K_t = K_{t+1}$ unless, during timestep t, an update to some state-action value occurs or the escape event A_K happens. Let M_{K_t} be the known state-action MDP and π_t be the current greedy policy, that is, for all states s, $\pi_t(s) = \mathrm{argmax}_a Q_t(s,a)$. Furthermore, assume $Q_t(s,a) \leq V_{\max}$ for all t and (s,a). Suppose that for any inputs ε and δ, with probability at least $1-\delta$, the following conditions hold for all states s, actions a, and timesteps t: (1) $V_t(s) \geq V^*(s) - \varepsilon$ (optimism), (2) $V_t(s) - V^{\pi_t}_{M_{K_t}}(s) \leq \varepsilon$ (accuracy), and (3) the total number of updates of action-value estimates plus the number of times the escape event from K_t, A_K, can occur is bounded by $\zeta(\varepsilon,\delta)$ (learning complexity). Then, when $\mathcal{A}(\varepsilon,\delta)$ is executed on any MDP M, it will follow a 4ε-optimal policy from its current state on all but*

$$O\left(\frac{V_{\max}\zeta(\varepsilon,\delta)}{\varepsilon(1-\gamma)} \ln\frac{1}{\delta} \ln\frac{1}{\varepsilon(1-\gamma)}\right)$$

timesteps, with probability at least $1-2\delta$.

- Estimated value never exceeds upper bound Vmax = R_{max} / (1-γ)

Probably (1-δ) Approximately (ε)

Theorem 10 *Let $\mathcal{A}(\varepsilon,\delta)$ be any greedy learning algorithm such that, for every timestep t, there exists a set K_t of state-action pairs that depends only on the agent's history up to timestep t. We assume that $K_t = K_{t+1}$ unless, during timestep t, an update to some state-action value occurs or the escape event A_K happens. Let M_{K_t} be the known state-action MDP and π_t be the current greedy policy, that is, for all states s, $\pi_t(s) = \mathrm{argmax}_a Q_t(s,a)$. Furthermore, assume $Q_t(s,a) \leq V_{\max}$ for all t and (s,a). Suppose that for $\boxed{\text{any inputs } \varepsilon \text{ and } \delta\text{, with probability at least } 1-\delta\text{,}}$ the following conditions hold for all states s, actions a, and timesteps t: (1) $V_t(s) \geq V^*(s) - \varepsilon$ (optimism), (2) $V_t(s) - V^{\pi_t}_{M_{K_t}}(s) \leq \varepsilon$ (accuracy), and (3) the total number of updates of action-value estimates plus the number of times the escape event from K_t, A_K, can occur is bounded by $\zeta(\varepsilon,\delta)$ (learning complexity). Then, when $\mathcal{A}(\varepsilon,\delta)$ is executed on any MDP M, it will follow a 4ε-optimal policy from its current state on all but*

$$O\left(\frac{V_{\max}\zeta(\varepsilon,\delta)}{\varepsilon(1-\gamma)} \ln\frac{1}{\delta} \ln\frac{1}{\varepsilon(1-\gamma)}\right)$$

timesteps, with probability at least $1-2\delta$.

- Specify how close want resulting policy to be to optimal
- Specify with what probability want bound on # of mistakes to hold

Assume that: Algorithm is Optimistic

Theorem 10 *Let $\mathcal{A}(\varepsilon,\delta)$ be any greedy learning algorithm such that, for every timestep t, there exists a set K_t of state-action pairs that depends only on the agent's history up to timestep t. We assume that $K_t = K_{t+1}$ unless, during timestep t, an update to some state-action value occurs or the escape event A_K happens. Let M_{K_t} be the known state-action MDP and π_t be the current greedy policy, that is, for all states s, $\pi_t(s) = \mathrm{argmax}_a Q_t(s,a)$. Furthermore, assume $Q_t(s,a) \leq V_{\max}$ for all t and (s,a). Suppose that for any inputs ε and δ, with probability at least $1-\delta$, the following conditions hold for all states s, actions a, and timesteps t: (1) $\boxed{V_t(s) \geq V^*(s) - \varepsilon \text{ (optimism)},}$ (2) $V_t(s) - V^{\pi_t}_{M_{K_t}}(s) \leq \varepsilon$ (accuracy), and (3) the total number of updates of action-value estimates plus the number of times the escape event from K_t, A_K, can occur is bounded by $\zeta(\varepsilon,\delta)$ (learning complexity). Then, when $\mathcal{A}(\varepsilon,\delta)$ is executed on any MDP M, it will follow a 4ε-optimal policy from its current state on all but*

$$O\left(\frac{V_{\max}\zeta(\varepsilon,\delta)}{\varepsilon(1-\gamma)}\ln\frac{1}{\delta}\ln\frac{1}{\varepsilon(1-\gamma)}\right)$$

timesteps, with probability at least $1-2\delta$.

- Algorithm's V_t and Q_t are always at least epsilon-optimistic wrt optimal V*
- Will values computed in R-max algorithm satisfy this?

Assume that: Algorithm is “Accurate”

Theorem 10 *Let $\mathcal{A}(\varepsilon, \delta)$ be any greedy learning algorithm such that, for every timestep t, there exists a set K_t of state-action pairs that depends only on the agent's history up to timestep t. We assume that $K_t = K_{t+1}$ unless, during timestep t, an update to some state-action value occurs or the escape event A_K happens. Let M_{K_t} be the known state-action MDP and π_t be the current greedy policy, that is, for all states s, $\pi_t(s) = \mathrm{argmax}_a Q_t(s,a)$. Furthermore, assume $Q_t(s,a) \le V_{\max}$ for all t and (s,a). Suppose that for any inputs ε and δ, with probability at least $1 - \delta$, the following conditions hold for all states s, actions a, and timesteps t: (1) $V_t(s) \ge V^*(s) - \varepsilon$ (optimism), (2) $\boxed{V_t(s) - V^{\pi_t}_{M_{K_t}}(s) \le \varepsilon \text{ (accuracy)}}$, and (3) the total number of updates of action-value estimates plus the*

- What would this mean for R-max?
- In R-max V_t is computed using following MDP M1
  - for (s,a) pairs in K_t: Use empirical estimate of transition and rewards
  - Else set to self loop with reward Rmax (means $Q(s,a) = R_{max} / (1-\gamma)$)

Assume that: Algorithm is "Accurate"

Theorem 10 *Let $\mathcal{A}(\varepsilon,\delta)$ be any greedy learning algorithm such that, for every timestep t, there exists a set K_t of state-action pairs that depends only on the agent's history up to timestep t. We assume that $K_t = K_{t+1}$ unless, during timestep t, an update to some state-action value occurs or the escape event A_K happens. Let M_{K_t} be the known state-action MDP and π_t be the current greedy policy, that is, for all states s, $\pi_t(s) = \mathrm{argmax}_a Q_t(s,a)$. Furthermore, assume $Q_t(s,a) \le V_{\max}$ for all t and (s,a). Suppose that for any inputs ε and δ, with probability at least $1-\delta$, the following conditions hold for all states s, actions a, and timesteps t: (1) $V_t(s) \ge V^*(s) - \varepsilon$ (optimism), (2) $\boxed{V_t(s) - V^{\pi_t}_{M_{K_t}}(s) \le \varepsilon \text{ (accuracy}}$), and (3) the total number of updates of action-value estimates plus the*

- What would this mean for R-max?
- In R-max V_t is computed using following MDP M1
  - for (s,a) pairs in K_t: Use empirical estimate of transition and rewards
  - Else set to self loop with reward Rmax (means $Q(s,a)= R_{max} / (1-\gamma)$)
- Recall M_{Kt} is defined as
  - For (s,a) pairs in K_t: Use true MDP transition and reward model
  - Else set to get value of $Q(s,a) = R_{max} / (1-\gamma)$
- This requires that both MDPs have near same computed value for pi for M1

Bounded Learning Complexity

Theorem 10 *Let $\mathcal{A}(\varepsilon, \delta)$ be any greedy learning algorithm such that, for every timestep t, there exists a set K_t of state-action pairs that depends only on the agent's history up to timestep t. We assume that $K_t = K_{t+1}$ unless, during timestep t, an update to some state-action value occurs or the escape event A_K happens. Let M_{K_t} be the known state-action MDP and π_t be the current greedy policy, that is, for all states s, $\pi_t(s) = \mathrm{argmax}_a Q_t(s,a)$. Furthermore, assume $Q_t(s,a) \leq V_{\max}$ for all t and (s,a). Suppose that for any inputs ε and δ, with probability at least $1-\delta$, the following conditions hold for all states s, actions a, and timesteps t: (1) $V_t(s) \geq V^*(s) - \varepsilon$ (optimism), (2) $V_t(s) - V^{\pi_t}_{M_{K_t}}(s) \leq \varepsilon$ (accuracy), and (3) the total number of updates of action-value estimates plus the number of times the escape event from K_t, A_K, can occur is bounded by $\zeta(\varepsilon, \delta)$ (learning complexity).*

- Most important: number of times a (s,a) pair can become known is bounded
- Somewhat intuitive: finite number of (s,a) pairs

Sufficient Condition for PAC Model-based RL

(see Strehl, Li, LIttman 2009, http://www.jmlr.org/papers/volume10/strehl09a/strehl09a.pdf)

Theorem 10 *Let $\mathcal{A}(\varepsilon,\delta)$ be any greedy learning algorithm such that, for every timestep t, there exists a set K_t of state-action pairs that depends only on the agent's history up to timestep t. We assume that $K_t = K_{t+1}$ unless, during timestep t, an update to some state-action value occurs or the escape event A_K happens. Let M_{K_t} be the known state-action MDP and π_t be the current greedy policy, that is, for all states s, $\pi_t(s) = \mathrm{argmax}_a Q_t(s,a)$. Furthermore, assume $Q_t(s,a) \leq V_{\max}$ for all t and (s,a). Suppose that for any inputs ε and δ, with probability at least $1-\delta$, the following conditions hold for all states s, actions a, and timesteps t: (1) $V_t(s) \geq V^*(s) - \varepsilon$ (optimism), (2) $V_t(s) - V^{\pi_t}_{M_{K_t}}(s) \leq \varepsilon$ (accuracy), and (3) the total number of updates of action-value estimates plus the number of times the escape event from K_t, A_K, can occur is bounded by $\zeta(\varepsilon,\delta)$ (learning complexity). Then, when $\mathcal{A}(\varepsilon,\delta)$ is executed on any MDP M, it will follow a 4ε-optimal policy from its current state on all but*

$$O\left(\frac{V_{\max}\zeta(\varepsilon,\delta)}{\varepsilon(1-\gamma)} \ln\frac{1}{\delta} \ln\frac{1}{\varepsilon(1-\gamma)}\right)$$

timesteps, with probability at least $1-2\delta$.

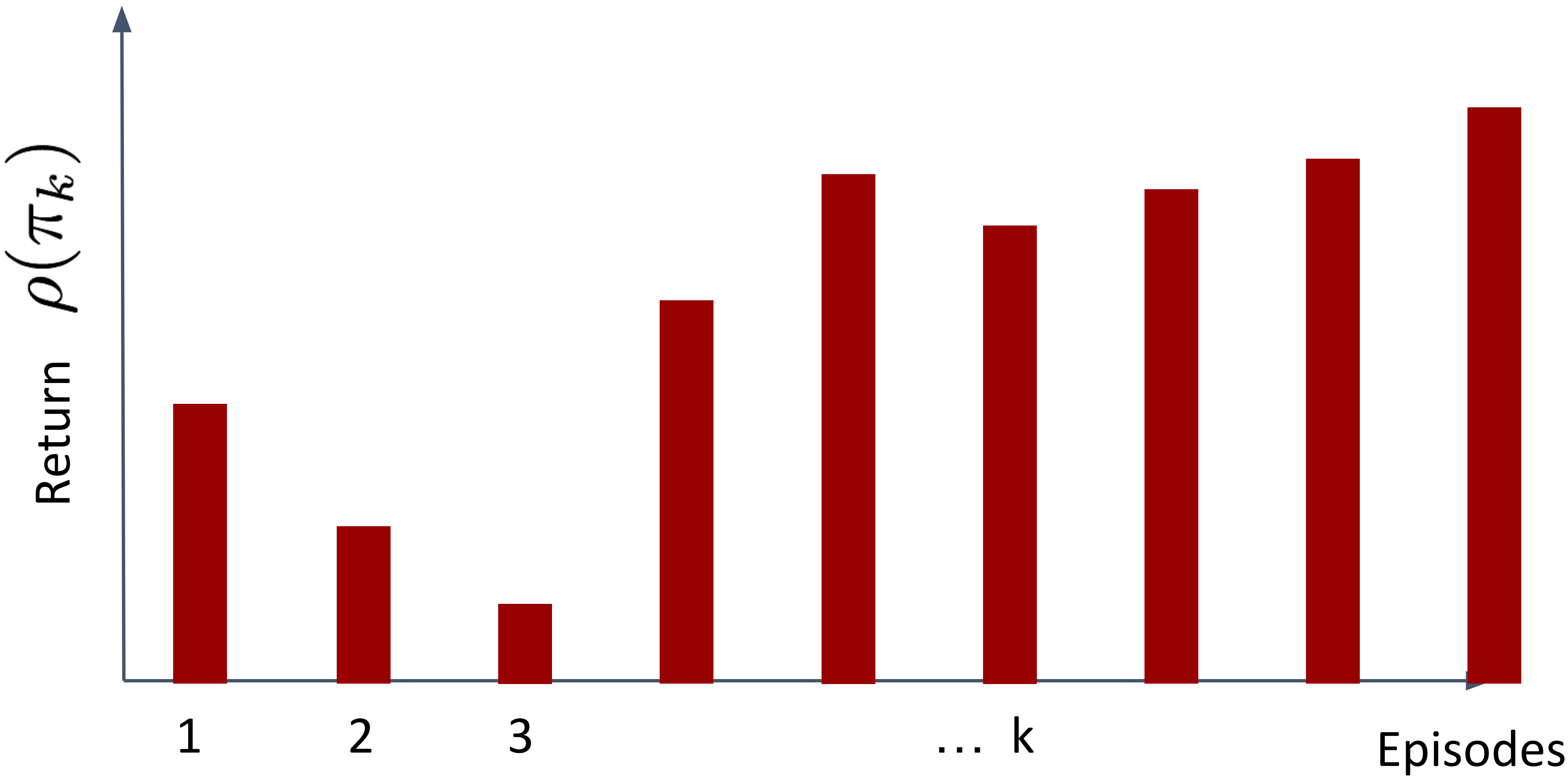
- If time: do proof on the board. Else see lecture notes for today's class

Optimism under Uncertainty

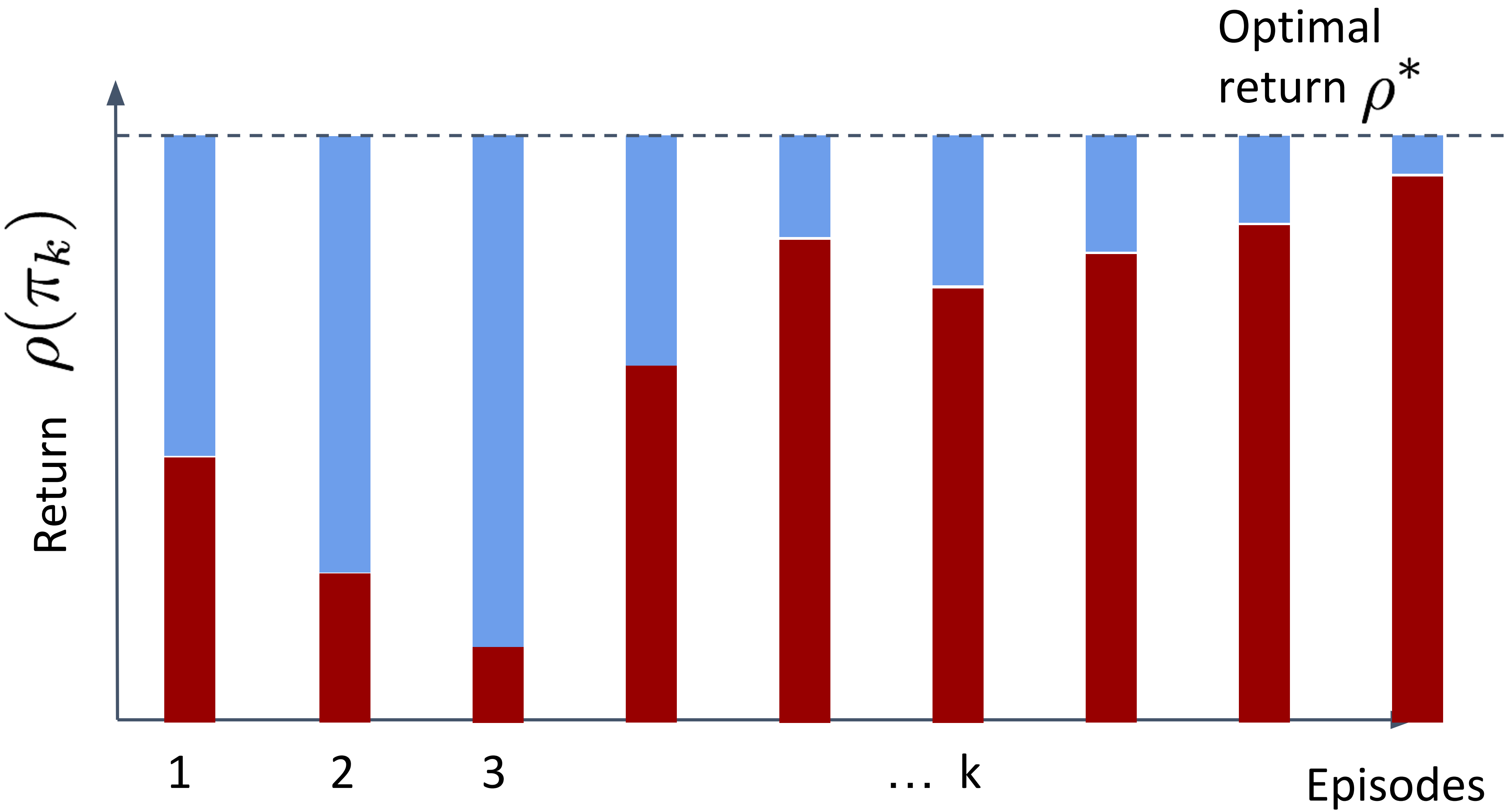
- Used in many algorithms that are PAC or regret
- Last lecture: UCRL2
  - Continuous representation of uncertainty
  - Confidence sets over model parameters
  - Regret bounds
- Today: R-max (Brafman and Tenneholtz)
  - Discrete representation of uncertainty
  - PAC bounds

Regret vs PAC vs ...?

- What choice of performance should we care about?
- For simplicity, consider **episodic** setting
- Return is the sum of rewards in an episode



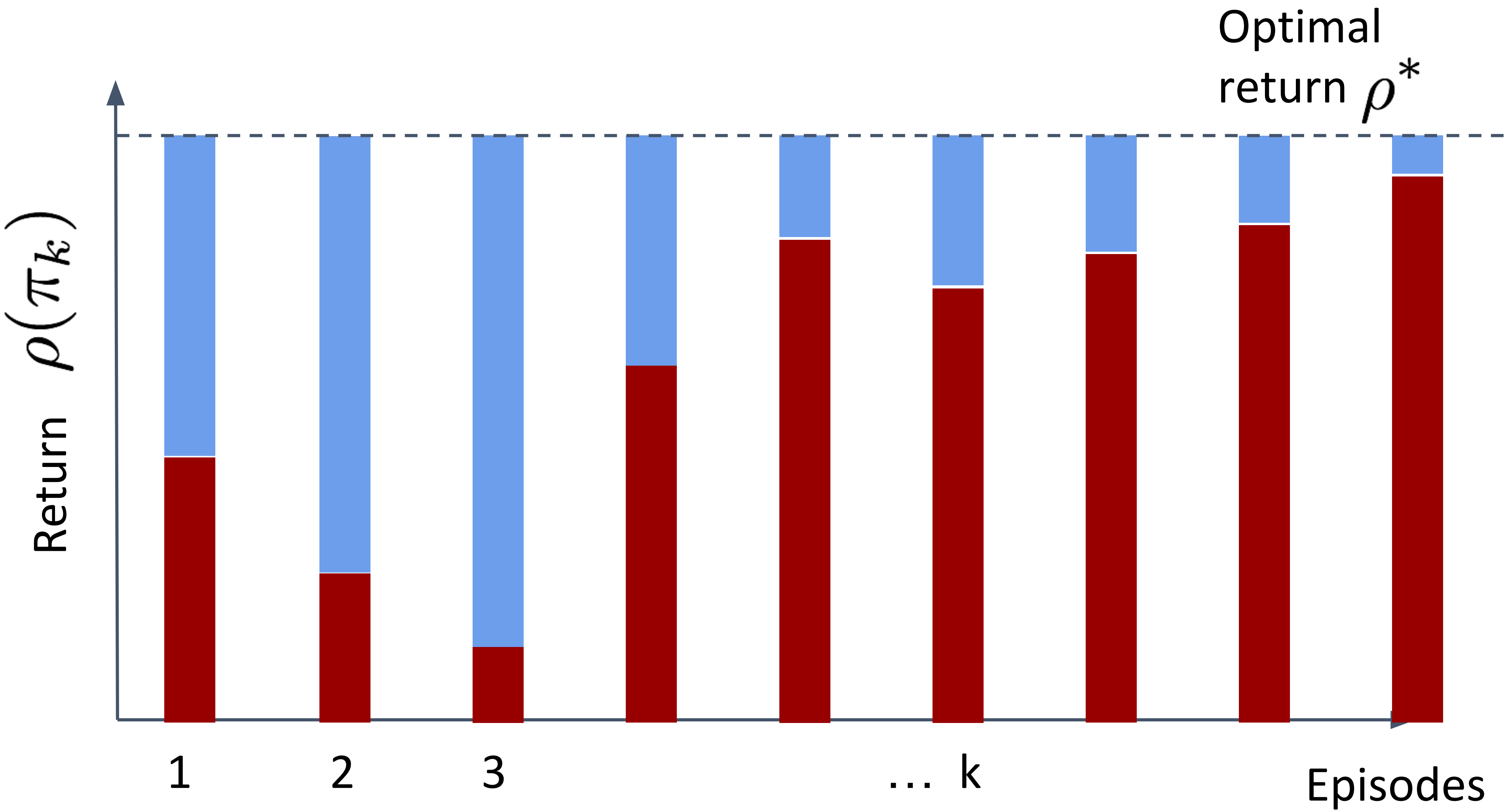
Regret Bounds



Regret Bounds

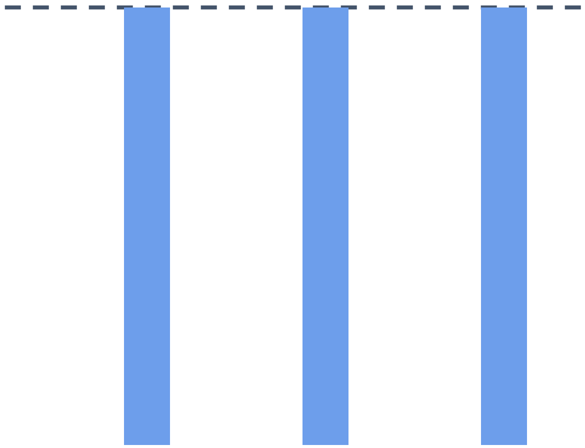
$$R(T) = T\rho^* - \sum_{k=1}^{T} \rho(\pi_k)$$

Guarantees bound expected regret $\mathbb{E}[R(T)]$

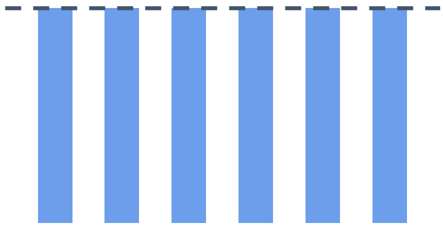


Expected Regret Limitations

- Algorithm only works in expectation
- No information on severity of mistakes

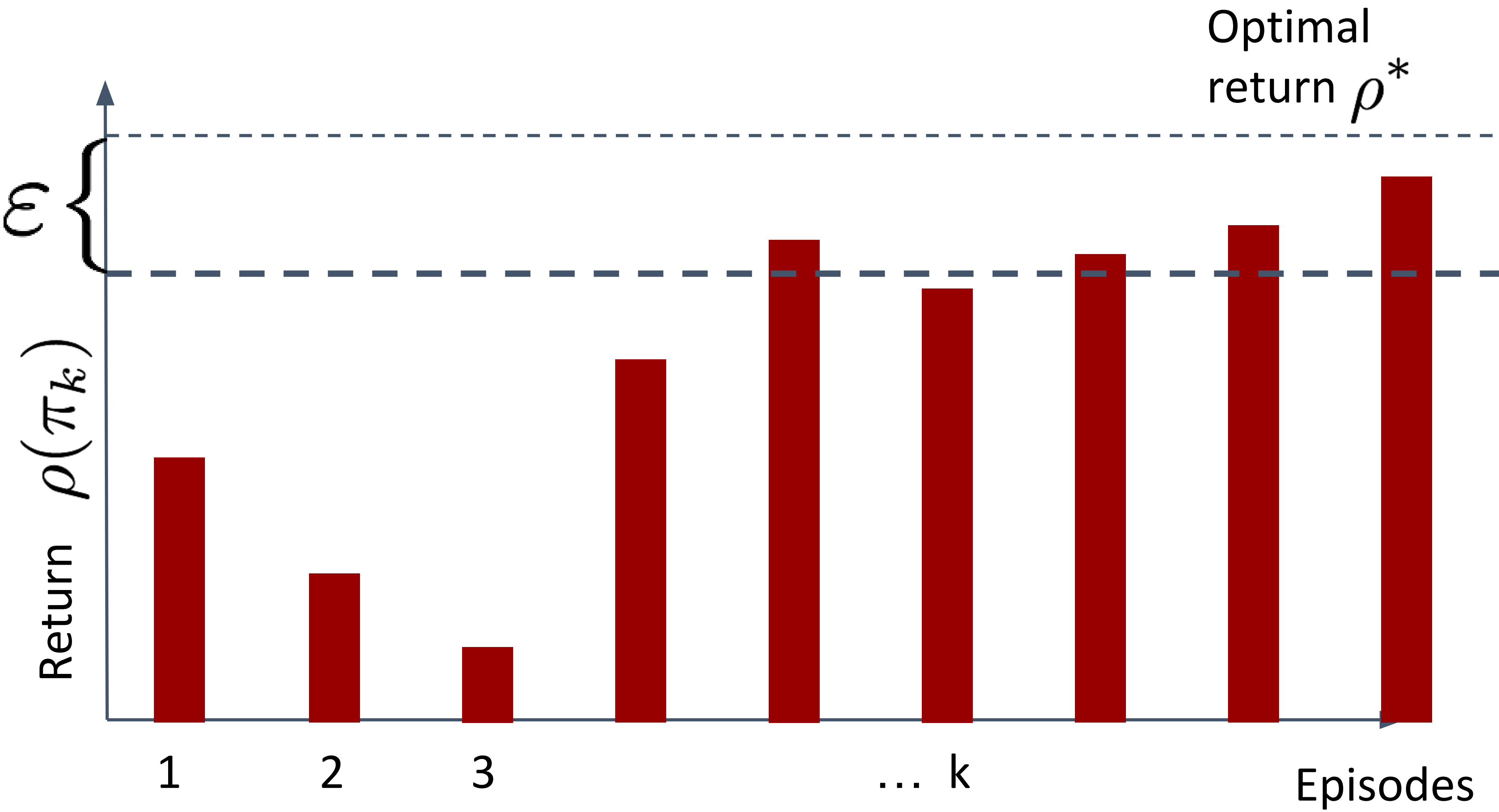


Few severely bad episodes
(Chronic severe pain)

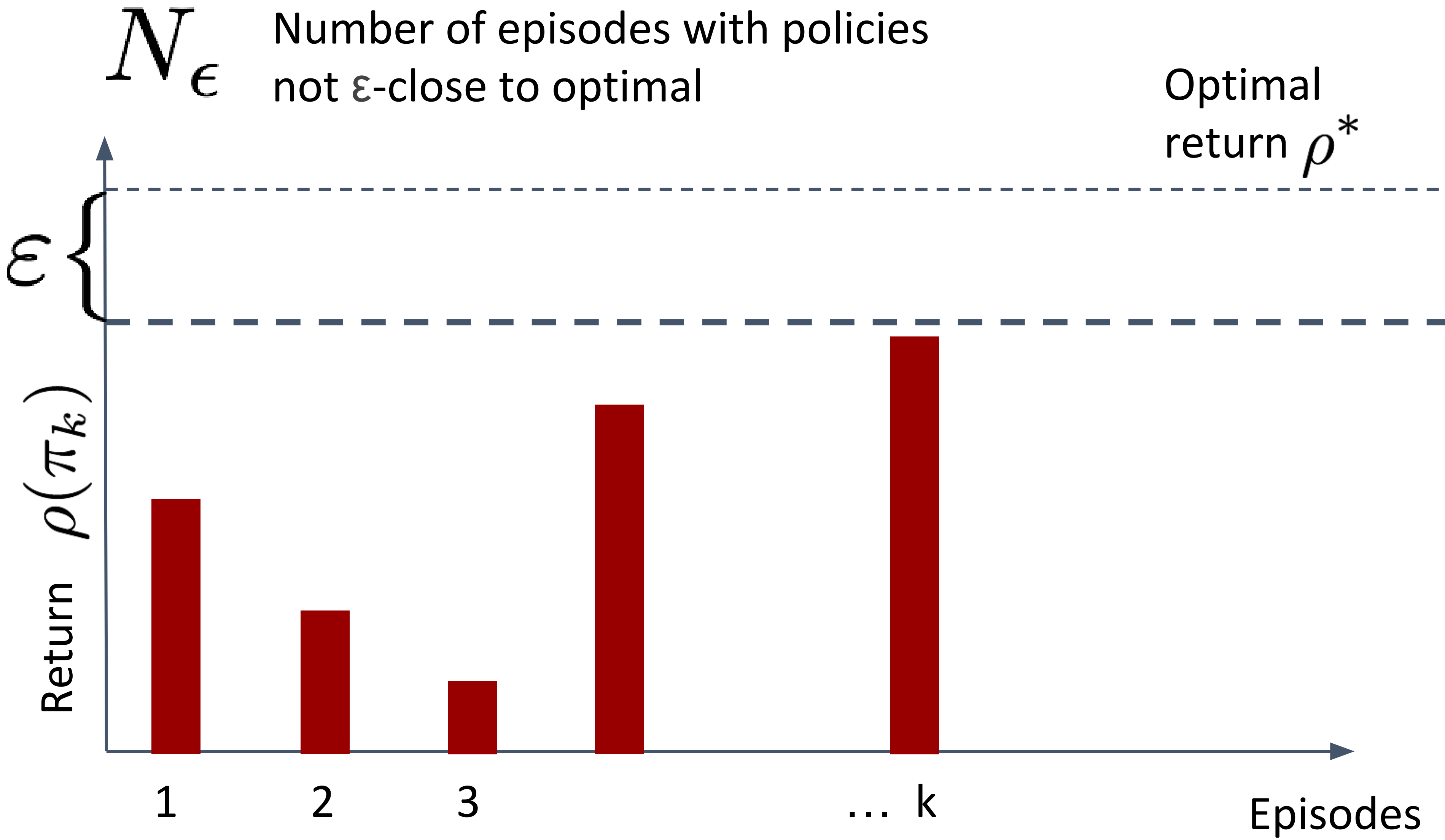


All episodes good but
not great (everyone has
a headache)

(ε,δ) - Probably Approximately Correct



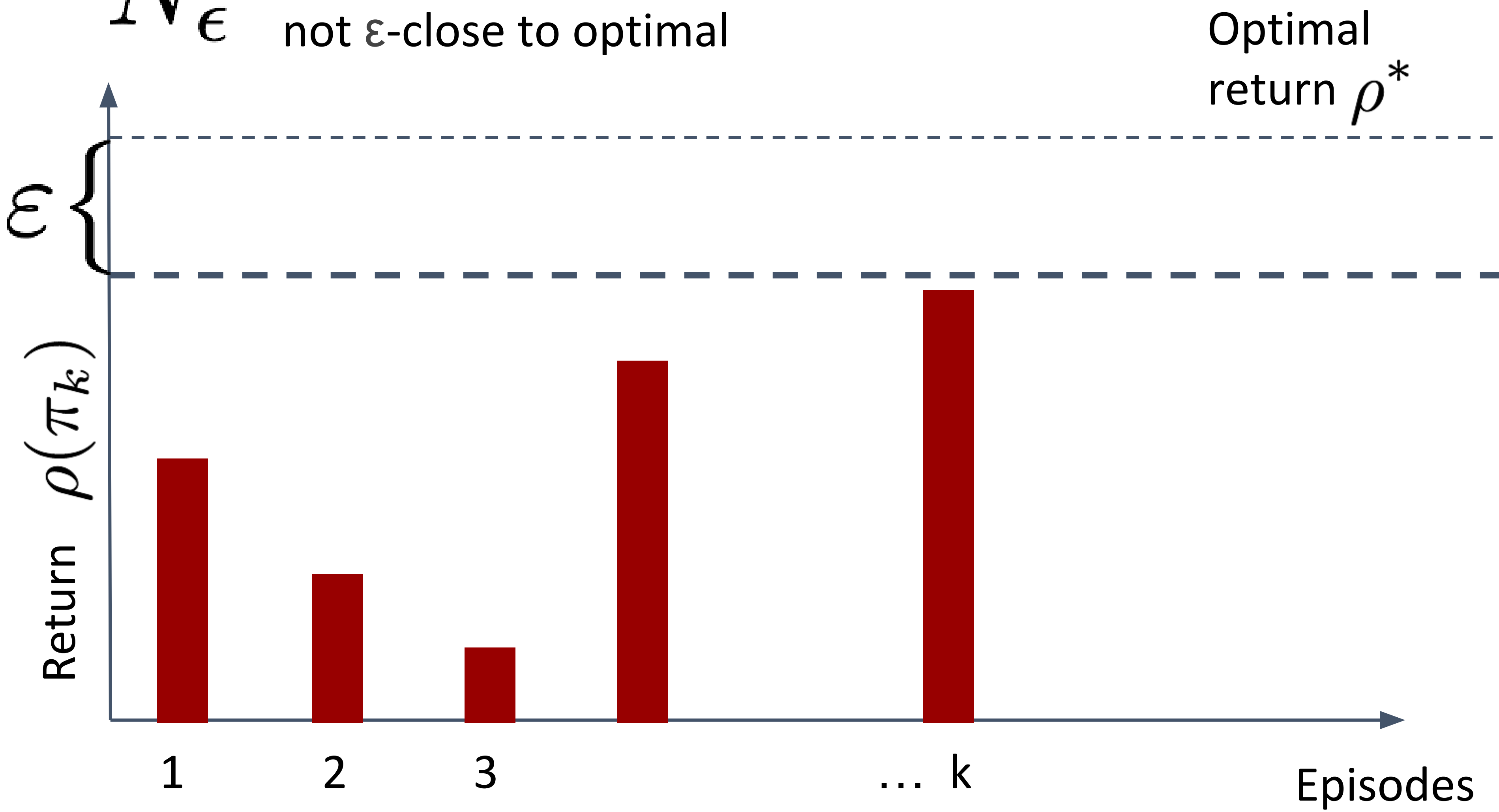
(ε,δ) - Probably Approximately Correct



(ε,δ) - Probably Approximately Correct

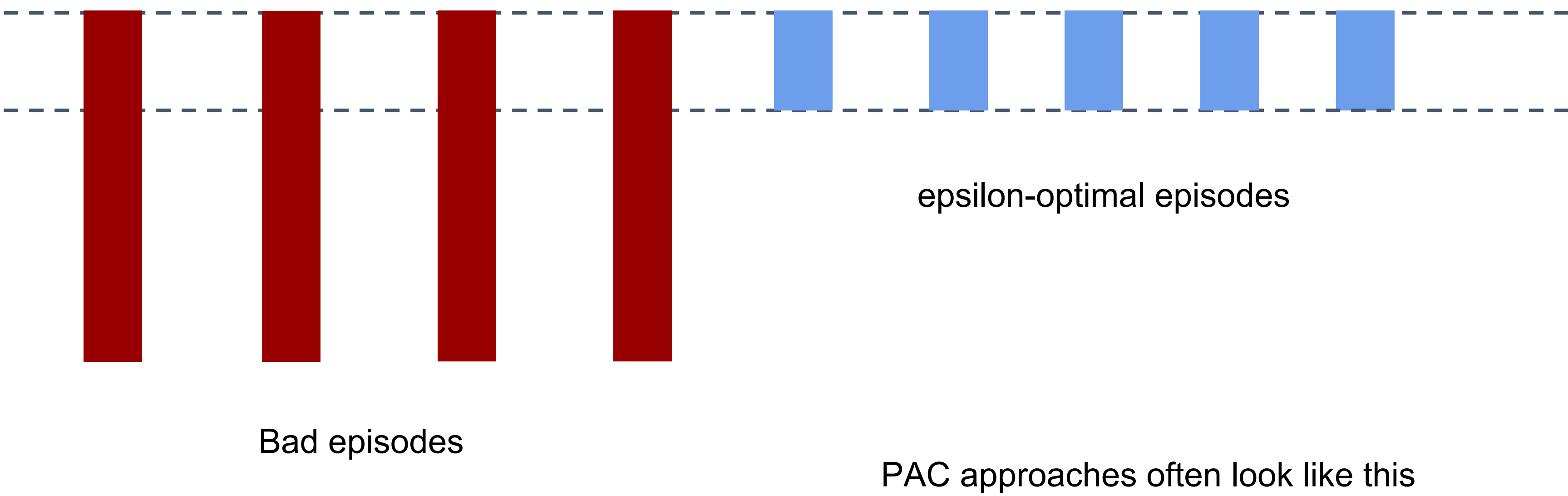
$$\mathbb{P}(N_\epsilon \leq F(S, A, H, \epsilon, \delta)) \geq 1 - \delta$$

N_ϵ Number of episodes with policies not ε-close to optimal



PAC Limitations

- Bound only on number of ε-suboptimal episodes, no guarantee of *how* bad they are
- Algorithm may not converge to optimal policy
- ε has to be determined a-priori

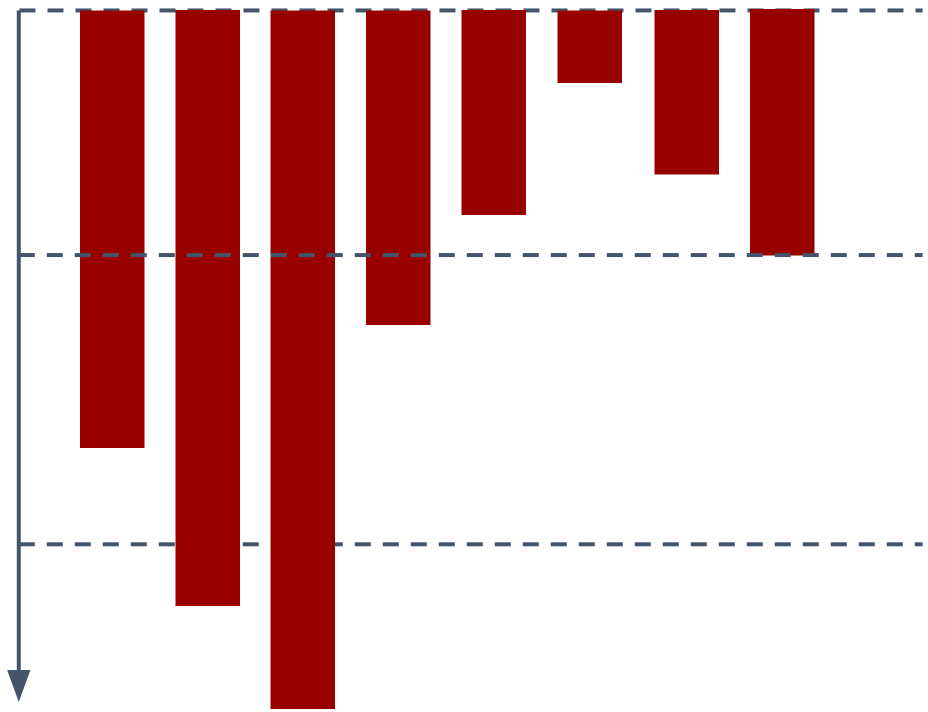


Uniform-PAC

(Dann, Lattimore, Brunskill, arxiv, 2017)

$$\mathbb{P}(\forall \varepsilon : N_\varepsilon \leq F(S, A, H, \varepsilon, \delta)) \geq 1 - \delta$$

bound on mistakes for any accuracy-level ε **jointly**



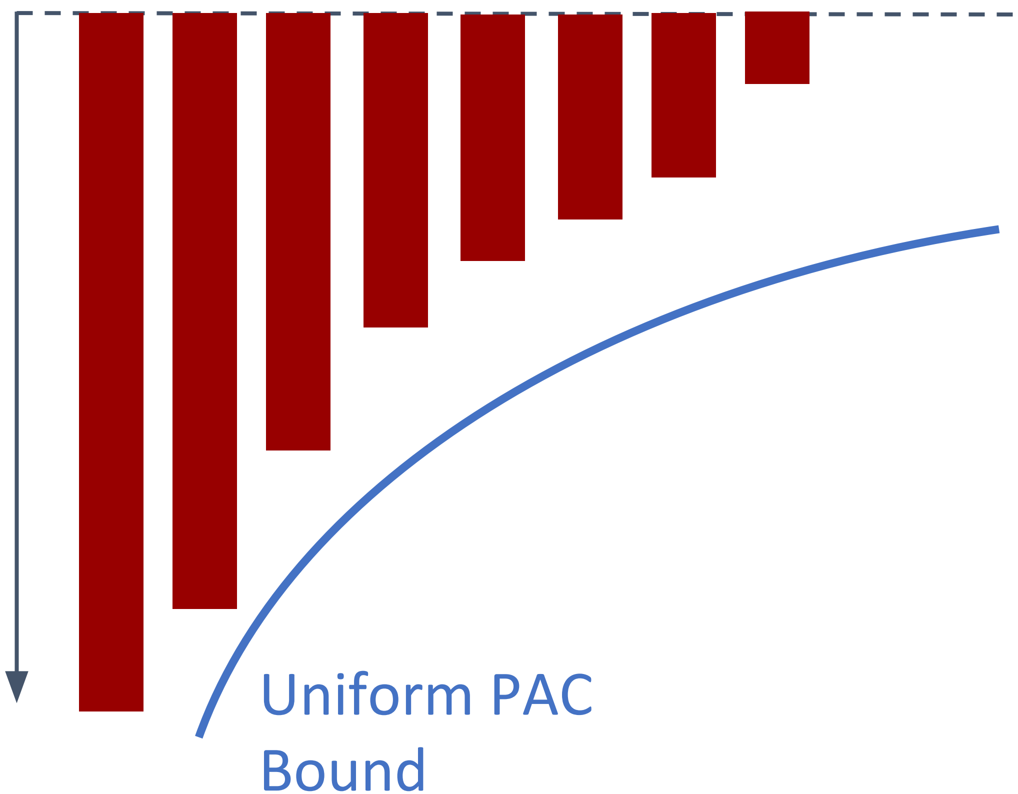
- Removes limitations listed including
- Algorithm converges to optimal policy
- No need to specify ε has to be determined a-priori

Uniform-PAC

(Dann, Lattimore, Brunskill, arxiv, 2017)

A δ-Uniform PAC-bound implies with prob. > 1 - δ :

- Convergence to π^*
- (ε,δ) - PAC $\forall\ \varepsilon$
- Regret bound $R(T)$

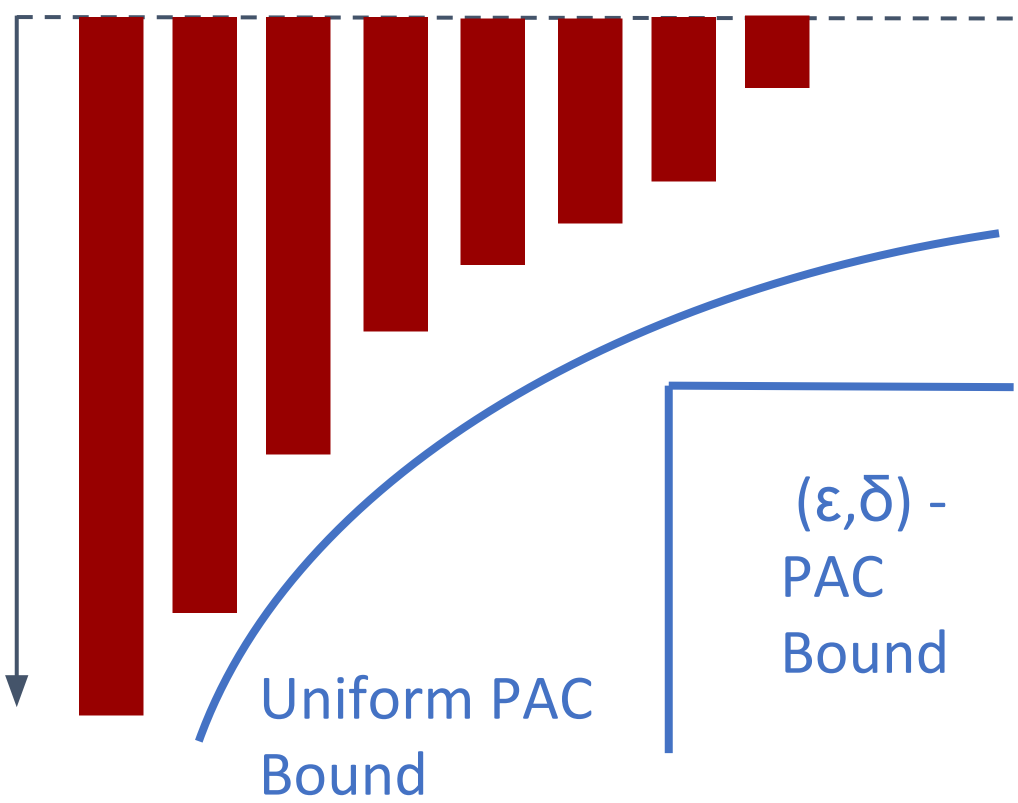


Uniform-PAC

(Dann, Lattimore, Brunskill, arxiv, 2017)

A δ-Uniform PAC-bound implies with prob. > 1 - δ :

- Convergence to π^*
- (ε,δ) - PAC $\forall\ \varepsilon$
- Regret bound $R(T)$



Summary

- Exploration is important
- Optimism under uncertainty can
  - Yield formal bounds on algorithm's performance
  - Have practical benefits
- Regret and PAC have some limitations, PAC-uniform is a new theoretical framework to get us closer to what we want in practice
- Still a large gap between bounds and practical performance

What You Should Understand

- Define 4 performance criteria and give examples where might prefer one over another
- Be able to implement at least 2 approaches to exploration